Byzantine-robust federated learning via credibility assessment on non-IID data

Kun Zhai, Qiang Ren, Junli Wang and Chungang Yan*

Key Laboratory of Embedded System and Service Computing (Tongji University), Ministry of Education, Shanghai 201804, China

* Correspondence: Email: yanchungang@tongji.edu.cn.

Abstract: Federated learning is a novel framework that enables resource-constrained edge devices to jointly learn a model, which solves the problem of data protection and data islands. However, standard federated learning is vulnerable to Byzantine attacks, which will cause the global model to be manipulated by the attacker or fail to converge. On non-iid data, the current methods are not effective in defending against Byzantine attacks. In this paper, we propose a Byzantine-robust framework for federated learning via credibility assessment on non-iid data (BRCA). Credibility assessment is designed to detect Byzantine attacks by combing adaptive anomaly detection model and data verification. Specially, an adaptive mechanism is incorporated into the anomaly detection model for the training and prediction of the model. Simultaneously, a unified update algorithm is given to guarantee that the global model has a consistent direction. On non-iid data, our experiments demonstrate that the BRCA is more robust to Byzantine attacks compared with conventional methods.

Keywords: Byzantine robust; federated learning; adaptive anomaly detection; non-iid; computer vision

1. Introduction

In recent years, the abundance of data generated from many distributed devices with the popularity of smartphones, wearable devices, intelligent home appliances, and autonomous driving. These data are usually concentrated in the data center for effective use. However, a crucial issue
arises that the concentrated data store causes leakage of personal privacy [1]. Simultaneously, as the computing power of these mobile devices increases, it is attractive to store data locally while completing related computing tasks. Federated learning is a distributed machine learning framework that allows multiple parties to collaboratively train a model without sharing raw data [2,3], which has attracted significant attention from industry and academia recently. [4] summarizes and discusses in the application of federated learning in big data and its future direction. Although federated learning has essential significance and advantages in protecting user privacy, it also faces many challenges.

First of all, due to the distributed nature of federated learning, it is vulnerable to Byzantine attacks. Notably, it has been shown that, with just one Byzantine client, the whole federated optimization algorithm can be compromised and fail to converge [5]. Especially when the training data is not independent and identically distributed (non-iid), the difficulty of defense against Byzantine attacks is increased and it is difficult to guarantee the convergence of the model [6].

Methods for defending against Byzantine attacks in federated learning have been extensively studied, including coordinate-wise trimmed mean [9], the coordinate-wise median [7,8], the geometric median [10,11], and distance-based methods Krum [12], BREA [6], Bulyan [5]. In addition to the above methods based on statistical knowledge, [14] proposes a new idea based on anomaly detection to complete the detection of Byzantine clients in the learning process. [13] discusses the challenges and future directions of federated learning in real-time scenarios in terms of cybersecurity.

The above methods can effectively defend against Byzantine attacks to some extent, but there are also some limitations. First, the methods based on statistical knowledge have high computational complexity, and also their defense abilities are weakened due to the non-iid data in federated learning. Second, for the anomaly detection algorithm [14], there is a premise that the detection model should be trained on the test data set. Obviously, the premise hypothesis cannot be realized in practical applications because it is difficult for us to get such a data set, which can cover almost all data distributions. Therefore, it is necessary for the anomaly detection model to get pre-training without relying on test dataset and update dynamically on non-iid data.

In this paper, we propose a new method that each client needs to share some data with the server, which makes a trade-off between client privacy and model performance. Unlike FedAvg [2], we use credibility score as the weight of model aggregation, not the sample size. The credibility score of each client is obtained by integrating the verification score and the detection score. The former is calculated by sharing data.

The main contributions of this paper are:

- We propose a new federated learning framework (BRCA) which combines credibility assessment and unified update. BRCA not only effectively defends against Byzantine attacks, but also reduces the impact of non-iid data on the aggregated global model.
- The credibility assessment combing anomaly detection and data verification effectively detects Byzantine attacks on non-iid data.
- By incorporating an adaptive mechanism and transfer learning into the anomaly detection model, the anomaly detection model can dynamically improve detection performance. Moreover, its pre-training no longer relies on the test data set.
- We customize four different data distributions for each data set, and explore the influence of data distribution on defense methods against Byzantine attacks.
2. Related work

FedAvg is firstly proposed in [2] as an aggregation algorithm for federated learning. The server updates the global model by a weighted average of the clients’ model updates, and the aggregation weight is determined based on its data sample size. Stich [15] and Woodworth et al. [16] analyze the convergence of FedAvg on strongly-convex smooth loss functions. However, they assume that the data is iid, which is not suitable for federated learning [17,18]. And Li et al. [19] makes the first convergence analysis of FedAvg when the data is non-iid. [20] uses clustering to improve federated learning in non-iid data. Regrettably, the ability of naive FedAvg is very weak to resist Byzantine attacks.

In the iterative process of federated aggregation, honest clients send the true model updates to the server, wishing to train a global model by consolidating their private data. However, Byzantine clients attempt to perturb the optimization process [21]. Byzantine attacks may be caused by some data corruption events in the computing or communication process such as software crashes, hardware failures and transmission errors. Simultaneously, they may also be caused by malicious clients through actively transmitting error information, in order to mislead the learning process [21].

Byzantine-robust federated learning has received increasing attention in recent years. Krum [12] is designed specially to defend Byzantine attacks in the federated learning. Krum generate the global model by a client’s model update whose distances to its neighbors is shortest. GeoMed [10] uses the geometric median which is a variant of the median from one dimension to multiple dimensions. Unlike the Krum, the GeoMed uses all client updates to generate a new global model, not just one client update.Trimmed Mean [9] proposes that each dimension of its global model is obtained by averaging the parameters of clients’ model updates in that dimension. But before calculating the average, the largest and smallest part of the parameters in that dimension are deleted, Xie et al. [22] and Mhamdi et al. [5] are all its variants. BREA [6] also considers the security of information transmission, but its defense method is still based on distance calculation. Zero [23] based on Watermark detection approach detect attacks such as malware and phishing attacks and cryptojacking. [24] surveys intrusion detection techniques in mobile cloud computing environment.

Table 1. The summary of the contributions and limitations of the related papers.

<table>
<thead>
<tr>
<th>Reference</th>
<th>Contributions</th>
<th>Limitations</th>
</tr>
</thead>
<tbody>
<tr>
<td>[12] [10] [9] [5] [22]</td>
<td>Krum, GeoMed and Trimmed Mean complete the Byzantine defense based on statistical knowledge. Easy to deploy applications.</td>
<td>The assumption is that the data of the clients is iid. High computational complexity.</td>
</tr>
<tr>
<td>[25]</td>
<td>The auto-encoder anomaly detection model is firstly applied to detect Byzantine attacks.</td>
<td>The pre-training of the anomaly detection model is completed on test dataset. The anomaly detection model is static.</td>
</tr>
<tr>
<td>[6]</td>
<td>Cryptography is used to protect the security of information transmitted between clients and server.</td>
<td>Defense against Byzantine attacks is still based on distance to find outliers, and had limited defenses capabilities.</td>
</tr>
</tbody>
</table>

All of the above defense methods based on statistical knowledge and distance are not effective in defending against Byzantine attacks in non-iid settings. Abnormal [25] uses an anomaly detection model to complete the detection of Byzantine attacks.

The concept of independent and identically distributed (iid) of data is clear, but there are many meanings of non-iid. In this work, we only consider label distribution skew \[17\]. The categories of samples may vary across clients. For example, in the face recognition task, each user generally has their face data; for mobile device, some users may use emojis that do not show up in others’ devices.

We summarize the contributions and limitations of the existing works in Table 1.

In this paper, we propose a method that combine credibility assessment and unified update to robust federated learning against Byzantine attacks on non-iid data.

3. Byzantine-robust federated learning on non-iid data

We utilize a federated setting that one server communicates with many clients. For the rest of the paper, we will use the following symbol definitions: \(A\) is the total client set, \(|A| = n\); \(S\) is the selected client set in every iteration, \(|S| = k\); among them, \(B\) is Byzantine client set, \(|B| = b\), and \(H\) is honest client set, \(|H| = h\). \(w_i^t\) is the model update sent by the client i to the server at round t, Byzantine attack rate \(\xi = \frac{b}{k} \cdot w_t^t\) is the global model at round t, \(D_P = \{D_1, ..., D_n\}\) is clients’ private data, \(D_s = \{D_s, ..., D_s\}\) is the clients’ shared data, and data-sharing rate \(\gamma = \frac{|D_s|}{|D_P| + |D_s|} \) (\(|\cdot|\) represents the sample size of the data set).

3.1. BRCA: Byzantine-robust federated learning via credibility assessment

In order to enhance the robustness of federated learning against Byzantines attacks on non-iid data, \(BRCA\) combines credibility assessment and unified update, Figure 1 depicts the architecture of \(BRCA\).

![Figure 1. The frame diagram of the BRCA.](image)

Before training, each client needs to share some private data to the server. In each iteration, the server randomly selects some clients and sends the latest global model to them. These clients use their
private data to train the model locally and send the model updates to the server. After receiving model updates, the server conducts a credibility assessment for each model update and calculates their credibility scores. Momentum is an effective measure to improve the ability of federated learning to resist Byzantine attacks [26]. So our aggregation Eq (1) is as follow:

\[ w^t + 1 = \alpha W^t + (1 - \alpha) \sum_{i \in S} r^t_i w^t_i \quad (1) \]

where \( r^t_i \) is the credibility score of client \( i \) at round \( t \) and \( \alpha (0 < \alpha < 1) \) is a decay factor. Last, unified update uses shared data to update the primary global model to get the new global model for this round.

Algorithm 1 is the description of BRCA, which contains Credibility Assessment in line 22, and Unified Update in line 28. The crucial of BRCA to defend against Byzantine attacks is credibility assessment. On non-iid data, the data distributions of different clients are immense, and it is difficult to judge whether the difference is caused by Byzantine attacks or the non-iid data. However, the model update of the honest client should have a positive effect on its private data, which is not affected by other clients. Simultaneously, anomaly detection model can effectively detect Byzantine attacks [25]. Thus, we combine the above two ideas to detect Byzantine attacks. In order to solve the shortcomings of the existing anomaly detection models, we propose an adaptive anomaly detection model. In this paper, the shared data is randomly selected by each client based on the sample category. Of course, other sampling methods could also be used, such as clustering. In addition, it must be pointed out that the shared data will only be used on the server, not on the clients. That effectively protect the clients’ privacy.

To summarize, BRCA has five steps. First: the server pre-train an anomaly detection model by source data and initialize a global model. Second: every client share little private data with the server. Three: every client download the newest global model from the server, and complete model updates by private data. Then, every client send the model update to the server. Four: the server update the global model and complete the adaptation of the anomaly detection model by model updates from clients. Five: the server update the primary global model with unified update, after that, the new global model is completed. Repeating steps three to five until the global model converges.

Our work is different from the recent state of the art. First, Krum, GeoMed and TrimmedMean are the representative methods based on geometric knowledge, but their premise is that the data of clients is dependent and identically distributed (iid). The hypothesis of our method is based on the actual application background of FL, aiming at non-iid data. Second, Abnormal is the first method to detect Byzantine attacks by auto-encoder anomaly detection model. However, the training of the anomaly detection model in the method is based on the test dataset and the abnormal detection model in the method is static. For both of the problems, our method has made improvement: 1) we pre-train the anomaly detection model with related but different source data without relying on the test dataset. 2) we introduce adaptive mechanism to the anomaly detection model, which help the detection model get update during federated iteration dynamically.

3.2. Credibility assessment

Algorithm 2(Credibility Assessment) is the key part of BRCA, which assigns a credibility score for each client model update. A Byzantine client would be given much lower credibility score than an honest client. To guarantee the accuracy of the credibility score, Credibility Assessment integrates adaptive anomaly detection model and data verification.
Algorithm 1: BRCA

**Input:** total clients $A$; total number of iterations $T$; learning rate $\eta_{\text{server}}, \eta_{\text{client}}, \eta_{\text{detection}}$;
Byzantine attack rate $\xi$; epoch $E_{\text{server}}, E_{\text{client}}$; initial global model $w^0$;
clients’ private data $D_P = \{D_1^P, \ldots, D_N^P\}$; clients’ shared data $D_s = \{D_1^S, \ldots, D_n^S\}$; initial anomaly detection model $\theta^0$; $\beta$; $\alpha$; $d$; $k$.

**Output:** global model $W^{T+1}$, anomaly detection model $\theta^{T+1}$.

1. $R = \emptyset$: the credibility score set.
2. $H = \emptyset$: the honest client set.
3. **Function** Add Attack($w$):
   - return $w$ attacked by Byzantine client.
4. **End Function**
5. **for each round** $t = 0$ to $T$ **do**
   6. **Clients:**
      - **for each** client $i \in S$ **do**
        7. **for each** epoch $e = 0$ to $E_{\text{client}}$ **do**
           8. $w_i^t = w^t - \eta_{\text{client}} \nabla_l(D_P^t, W^t)$, where $l$ is the loss function.
        9. **end**
      10. **if** $i \in \text{top } \xi \% \text{ of } S$ **then**
        11. $W_i^t = \text{Add Attack}(W_i^t)$
        12. **end**
      13. **send** $W_i^t$ **back to the server**
   14. **end**
   15. **Server:**
      16. **sample** $S \subseteq A$ **randomly**
      17. **broadcast** latest global model $w^t$ **to each client** $j \in S$
      18. **receive** model updates from clients $Q = \{W_1^t, \ldots, W_J^t, \ldots, W_k^t\}$, client $j \in S$
      19. $R, H, \theta^{T+1} = \text{Credibility Assessment} (Q, D_s, \theta^t, \beta, S, \eta_{\text{detection}}, d, k)$
      20. **if** $H > \frac{1}{2} k$ **then**
         21. $w^{t+1} = \alpha w^t + (1 - \alpha) \sum_{j \in S} r_j^t \ast w_j^t$, $r_j^t \in R$
      22. **else**
      23. $w^{t+1} = w^t$
      24. **end**
      25. $w^{t+1} = \text{Unified Update} (w^{t+1}, D_s, E_{\text{server}}, \eta_{\text{server}}, H)$
      26. **end**
   27. **return** global model $W^{t+1}$, anomaly detection model $\theta^{T+1}$.
In Algorithm 2, line 4 is the data verification, which calculates the verification score $f_i$ for the model update of client $i$. And line 5 is the get-anomaly-score() of the adaptive anomaly detection model, which calculates detection score $e_i$. Subsequently, the credibility $r_i$ of the model update is $r_i = \beta e_i + (1 - \beta)f_i$, $R = \{r_1, ..., r_l, ..., r_k\}$, client $i \in S$. The make-adaption () in line 24 implements the adaption of the anomaly detection model.

In this paper, we judge the model update with a credibility score lower than the mean of $R$ as a Byzantine attack, and set its credibility score as zero. Finally, normalizing the scores to get the final credibility scores.

3.2.1 Adaptive anomaly detection model

In the training process, we cannot predict the type of attacks, but we can estimate the model update of the honest client. Therefore, we can adopt a one-class classification algorithm to build the anomaly detection model with normal model updates. Such technique will learn the distribution boundary of the model updates to determine whether the new sample is abnormal. Auto-encoder is an effective one-class learning model for detecting anomalies, especially for high-dimensional data [27].

In practical applications, we cannot get the target data to complete the pre-training of our anomaly detection model. Therefore, the initialized anomaly detection model will be pre-trained on the source data with the idea of transfer learning.

At round $t$, the detection score $e_i^t$ of client $i$:

$$e_i^t = \exp \left( \frac{Mse(C_i^t - \beta(c_i^t)) - \mu(E)}{\sigma(E)} \right)$$  \hspace{1cm} (2)

Our anomaly detection model is different from the one in Abnormal: 1) Abnormal uses the test set of the data set to train the anomaly detection model. Although the detection model obtained can complete the detection task very well, in most cases the test data set is not available. Therefore, based on the idea of transfer learning, we complete the pre-training of the anomaly detection model in the source domain. 2) Abnormal’s anomaly detection model will not be updated after training on the test set. We think this is unreasonable, because the test set is only a tiny part of the overall data. Using a small part of the training data to detect most of the remaining data, and the result may not be accurate enough. Therefore, pre-training of the anomaly detection model is completed in the source domain. Then we use the data of the target domain to fine-tune it in the iterative process to update the anomaly detection model dynamically, as make-adaption shown in Algorithm 3.
Algorithm 2: Credibility Assessment

Input: local model updates $Q$; clients’ shared data $D_s = \{D_1^s, ..., D_n^s\}$; anomaly detection model $\theta^t$; $\beta$; selected clients $S$; $\eta_{\text{detection}}$, $d$, $k$

Output: credibility score of clients $R$; honest client set $H$; anomaly detection model $\theta^{t+1}$

$R = \emptyset$: credibility score set; $H = \emptyset$: the honest client set; $sum = 0$;

$C = \{C_1^t, ..., C_i^t, ..., C_k^t\}$, client $i \in S$, $c_i^t$ is the weight of the last convolutional layer of $W_i^t$

for each client $i \in S$ do
  \[ Data \text{ Verification: compute } f_i^t \text{ with } W_i^t \text{ and } D_i^t, \text{ client } i \in S \text{ base on equation 3 and equation 4} \]
  \[ e_i^t = \text{AADM. get-anomaly-score } (\theta^t, C) \]
end

$sum_e = \sum_{\text{client } i \in S} e_i^t$; $sum_f = \sum_{\text{client } i \in S} f_i^t$

for each client $i \in S$ do
  \[ e_i^t = e_i^t / sum_e ; f_i^t = f_i^t / sum_f \]
  \[ r_i^t = \beta e_i^t + (1 - \beta) f_i^t ; R = R \cup \{r_i^t\} \]
end

$M(R)$ is the mean of $R$

for each $r_i^t \in R$ do
  if $r_i^t < M(R)$ then
    \[ r_i^t = 0 \]
  else
    \[ H = H \cup \{i\} \]
end

$sum + = r_i^t$

for each $r_i^t \in R$ do
  \[ r_i^t = r_i^t / sum \]
end

$\theta^{t+1} = \text{AADM. make-adaption } (H, \theta^t, \eta_{\text{detection}}, C, d, k)$

return $R$, $H$, $\theta^{t+1}$.

3.2.2 Data verification

The non-iid of client data increases the difficulty of Byzantine defense. However, the performance of the updated model of each client on its shared data is not affected by other clients, which can be effectively solved this problem. Therefore, we use the clients’ shared data $\{D_s = D_1^s, ..., D_i^s, ..., D_k^s\}$
**Algorithm 3:** AADM adaptive anomaly detection model

**Input:** anomaly detection model \( \theta^t \); weights of the last convolutional layer of the local model \( C \); \( \eta_{\text{detection}} \); credibility score \( R \); honest client set \( H;d;k \)

**Output:** updated anomaly detection model \( \theta^{t+1} \)

1. **Function** get-anomaly-score \((\theta^t, C^t_i)\):
   2. compute \( e^t_i \) with \( \theta^t \) and \( c^t_i \), client \( i \in S \) based on Eq (2)
   3. return \( e^t_i \).

4. **Function** make-adaption \((H, \theta^t, \eta_{\text{detection}}, C, d)\):
   5. if \(|H| < 1/2 k\) then
      6. \( \theta^{t+1} = \theta^t \)
   7. else
      8. \( H_0 \) is a subset of \( H \) obtained by removing the clients whose credibility score is in the largest and smallest \( d \) fraction.
      9. for client \( i \in H_0 \) do
         10. \( \theta^{t+1} = \theta^t - \eta_{\text{detection}} \nabla l(\theta^t, c^t_i) \)
      11. end
   12. return updated anomaly detection model \( \theta^{t+1} \)

End Function

---

Client \( i \in S \) to calculate the verification score of their updated model:

\[
f^t_i = \left( \exp\left(\frac{l^t_i - \mu(L)}{\sigma(L)}\right) \right)^{-2}
\]  

(3)

where \( l^t_i \) is loss of client \( i \) calculated on model \( w^t_i \) using the shared data \( D^s_i \) at round \( t \):

\[
l^t_i = \frac{1}{|D^s_t|} \sum_{j=0}^{|D^s_t|} l(D^{s(j)}_i, W^t_i)
\]  

(4)

where \( D^{s(j)}_i \) is the \( j \)-th sample of \( D^s_i \) and \( \mu(L), \sigma(L) \) are the mean and variance of set \( L = \{l_1, ..., l_k\} \) respectively.

3.3. Unified update

After getting the credibility score \( r^k_i \) in Algorithm 2 with the anomaly score \( e^k_i \) and the verification score \( f^k_i \), we can complete the aggregation of the clients’ local model updates in Eq (1) and get a preliminary updated global model. However, due to the non-iid of client data, the knowledge learned by the local model of each client is limited, and the model differences between two clients are also significant. Therefore, to solve the problem that the preliminary aggregation model lacks a clear
and consistent goal, we introduce an additional **unified update** procedure with shared data on server, details can be seen in Algorithm 4.

Because the data used for the **unified update** is composed of each client’s data, it can more comprehensively cover the distribution of the overall data. The goal and direction of the **unified update** are based on the overall situation and will not tend to individual data distribution.

Algorithm 4: Unified update

<table>
<thead>
<tr>
<th>Input:</th>
<th>global model $w^{t+1}$; clients’ shared data $D_s = {D_1^s, ..., D_n^s}$; $E_{server}$; $\eta_{server}$; honest client set $H$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output:</td>
<td>global model $w^{t+1}$.</td>
</tr>
</tbody>
</table>

1. **for each epoch** $e = 0$ to $E_{server}$ **do**
   2. **for** $i \in H$ **do**
      3. $w^{t+1} = w^{t+1} - \eta_{server} \nabla l(D_i^s, w^{t+1})$
   4. **end**
5. **end**
6. **return** global model $w^{t+1}$.

4. Experiments

To verify the effectiveness of BRCA, we structure the client’s data into varying degrees of non-iid, and explore the impact of different amounts of shared data on the global model. At the same time, we also compare the performance of our anomaly detection model with the Abnormal’s and explore the necessity of unified update.

4.1. Experimental steup

4.1.1. Datasets

Mnist and Cifar10 are the two most commonly used public data sets in image classification, and most of the benchmark methods in our work also use these two data sets for experiments. Using these two data sets, it is easier to compare with other existing methods.

We do the experiments on Mnist and Cifar10, and customize four different data distributions: (a) non-iid-1: each client only has one class of data. (b) non-iid-2: each client has 2 classes of data. (c) non-iid-3: each client has 5 classes of data. (d) iid: each client has 10 classes of data.

For Mnist, using 100 clients and four data distributions: (a) non-iid-1: each class of data in the training dataset is divided into 10 pieces, and each client selects one piece as its private data. (b) non-iid-2: each class of data in the training dataset is divided into 20 pieces, and each client selects 2 pieces of different classes of the data. (c) non-iid-3 each class of data in the training dataset is divided into 50 pieces, and each client selects 5 pieces of different classes of the data. (d) iid: each class of data in the training dataset is divided into 100 pieces, and each client selects 10 pieces of different classes of the data. As for the source domains used for the pre-training of the anomaly detection model, we randomly select 20,000 lowercase letters in the Nist dataset.

For Cifar10, there are 10 clients and the configuration of four data distributions is similar to that of the Mnist. We select some classes of data in Cifar100 as source domain, which are as follows: lamp
(number:40), lawn mower (41), lobster (45), man (46), forest (47), mountain (49), girl (35), Snake (78), Rose (70) and Tao (68), these samples do not exit in Cifar10.

4.1.2. Models

We use logistic regression on Mnist dataset. $\eta_{server} = 0.1$, $\eta_{client} = 0.1$, $\eta_{detection} = 0.02$, $E_{client}= 5$, $E_{server} = 1$, $n = 100$, $k = 30$, $\xi = 20\%$. Two convolution layers and three fully connected layer on Cifar10, $\eta_{server} = 0.05$, $\eta_{client} = 0.05$, $\eta_{detection} = 0.002$, $E_{client} = 10$, $E_{server} = 10$, $n = 10$, $k = 10$, $\xi = 20\%$. The structure of models are the same as [10].

4.1.3. Benchmark byzantine attacks

Same-value attacks: A Byzantine client $i$ sends the model update $\omega_i = c1$ to the server ($1$ is all-ones vectors, $c$ is a constant), we set $c = 5$. Sign-flipping attacks: In this scenario, each client $i$ computes its true model update $\omega_i$, then Byzantine clients send $\omega_i = a\omega_i$ ($a < 0$) to the server, we set $a = -5$. Gaussian attacks: Byzantine clients add Gaussian noise to all the dimensions of the model update $\omega_i = \omega_i + \epsilon$, where $s$ follows Gaussian distribution $N (0, g^2)$ where $g$ is the variance, we set $g = 0.3$.

4.1.4. Benchmark defense methods

Defenses: Krum, GeoMed, Trimmed Mean, Abnormal and No Defense. No Defense does not use any defense methods.

4.2. Result and discussion

4.2.1. Impact of shared data rate

In the first experiment, we test the influence of the shared data rate $\gamma$ in our algorithm, and do the experiment with the data distribution of non-iid-2. We implement it on five different values [1, 3, 5, 7 and 10%]. Figures 2 and 3 are the accuracy and loss for Cifar10. It is found that: 1) In all cases of Byzantine attacks, our algorithm is superior to the three benchmark methods. 2) Only 1% of the data shared by the client can significantly improve the performance of the global model. For three Byzantine attacks, Krum, GeoMed, Trimmed Mean, No Defense are all unable to converge. This also shows that when the model is complex, such methods would be less able to resist Byzantine attacks.

With the increase in the client data sharing ratio, the performance of the global model has become lower. When the client shares the data ratio from 1 to 10%, the average growth rate with the three Byzantine attacks are: $1.8 \rightarrow 1.41 \rightarrow 0.97 \rightarrow 0.92\%$. The clients only share one percent of the data, and the performance of the global model can be greatly improved.

Figure 4 clearly demonstrates the impact of different shared data rates on the loss value of the global model on Cifar10.
Figure 2. The Accuracy of Cifar10. Byzantine attack types from (a) to (c) are as follows: Same value, Sign flipping and Gaussian noisy. Six defense methods are adopted for each type of attack, in order: No defense, Krum, GeoMed, Trimmed Mean, Abnormal and BRCA. For Ours, there are five different shared data rate (1, 3, 5, 7 and 10%), which correspond accordingly: BRCA 1, BRCA 3, BRCA 5, BRCA 7, BRCA 10.

Figure 3. The loss of Cifar10. The legends are the same as Figure 2.

Figure 4. The loss of BRCA on Cifar10 with five different shared rate.
4.2.2. Performance of anomaly detection model

In this part, the purposes of our experiment are: 1) Compare anomaly detection model between ours and Abnormal. 2) Explore the robustness of the anomaly detection model to data that are non-iid. The shared data rate $\gamma$ is 5%, Sections 4.2.3 and 4.2.4 are the same.

In order to compare the detection performance of the anomaly detection model against Byzantine attacks between BRCA and Abnormal, we use the cross-entropy loss as the evaluation metric which is calculated by the detection score. Firstly, we get detection scores $E = \{e_1, ..., e_i, ..., e_k\}$ based on model update $\omega_i$ and $\theta$, client $i \in S$. Then, we set $P = \text{Sigmoid}(E - \mu(E))$ represents the probability that the client is honest and $1 - P$ is the probability that the client is Byzantine. Lastly, we use $P$ and true label $Y (y^i = 0, i \in B \text{ and } y^j = 1, j \in H)$ to calculate the cross-entropy loss $l = \sum_{i=1}^{k} y_i \ln(P_i)$

![Figure 5](image1.png)

**Figure 5.** the cross-entropy loss of our and Abnormal anomaly detection model, on Cifar10 with non-iid-2. (a)–(c) are the performance for three Byzantine-attacks.

![Figure 6](image2.png)

**Figure 6.** (a)–(c) are our anomaly detection model’s performance on four different data distribution (iid, non-iid-1, non-iid-2, non-iid-3) against Byzantine attacks (Gaussian noisy, sign flipping, same value).

Figure 5(a)–(c) compare the loss of the anomaly detection model between BRCA and the Abnormal. From the figures, we can see that our model has a greater loss than Abnormal in the initial stage, mainly due to the pre-training of the anomaly detection model using the transfer learning.
initial pre-trained anomaly detection model cannot be used well in the target domain. As the adaptation progress, the loss of our model becomes decreases and gradually outperforms the Abnormal. Although Abnormal has a low loss in the initial stage, as the training progresses, the loss gradually increases, and the detection ability becomes degenerate.

Figure 6(a)–(c) show the influence of different data distributions on our detection model. For different data distributions, the detection ability of the model is different, but it is worth pointing out that: as the degree of non-iid of the data increases, the detection ability of the model also increases.

4.2.3. Impact of unified update

In this part, we study the impact of the unified update on the global model. Figure 7 shows the accuracy of the global model with and without unified update on Cifar10.

From non-iid-1 to iid, the improvement of the global model’s accuracy by unified update is as follows: 35.1→13.6→4.7→2.3% (Same value), 34.8→10.5→3.0→3.1% (Gaussian noisy), 24.9→9.9→2.8→3.0% (Sign flipping). Combined with Figure 7, it can be clearly found that the more simple the client data is, the more obvious the unified update will be to the improvement of the global model.

When the data is non-iid, the directions of the model updates between clients are different. The higher the degree of non-iid of data, the more significant the difference. The global model obtained by weighted aggregation does not fit well with the global data. Unified update on the shared data can effectively integrate the model updates of multiple clients, giving the global model a consistent direction.

Therefore, it is necessary to implement a unified update to the primary aggregation model when data is non-iid.

![Figure 7. The accuracy of BRCA and BRCA No on Cifar10. BRAC No is based on BRCA with unified update removed.](image)

(a) Same value  (b) Sign flipping  (c) Gaussian noisy

4.2.4. Impact of non-iid

Tables 2 and 3 show the accuracy and loss of each defense method under different data distributions on Cifar10. It can be seen that our method is the best, and the performance is relatively
stable for different data distributions. The higher the degree of non-iid of data, the more single the data of each client, the lower the performance of the defense method.

**Table 2.** The accuracy of the six defenses under four different data distributions on Cifar10, against three attacks.

<table>
<thead>
<tr>
<th>Attacks</th>
<th>Defenses</th>
<th>No</th>
<th>Krum</th>
<th>GeoMed</th>
<th>Abnormal</th>
<th>TrimmedMean</th>
<th>BRCA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Same value</td>
<td>Non-iid-1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.178</td>
<td>0.1</td>
<td>0.529</td>
</tr>
<tr>
<td></td>
<td>Non-iid-2</td>
<td>0.101</td>
<td>0.207</td>
<td>0.205</td>
<td>0.480</td>
<td>0.1</td>
<td>0.619</td>
</tr>
<tr>
<td></td>
<td>Non-iid-3</td>
<td>0.1</td>
<td>0.398</td>
<td>0.398</td>
<td>0.634</td>
<td>0.1</td>
<td>0.691</td>
</tr>
<tr>
<td></td>
<td>iid</td>
<td>0.098</td>
<td>0.696</td>
<td>0.705</td>
<td>0.698</td>
<td>0.101</td>
<td>0.713</td>
</tr>
<tr>
<td>Gaussian noisy</td>
<td>Non-iid-1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.178</td>
<td>0.1</td>
<td>0.529</td>
</tr>
<tr>
<td></td>
<td>Non-iid-2</td>
<td>0.191</td>
<td>0.204</td>
<td>0.205</td>
<td>0.513</td>
<td>0.059</td>
<td>0.623</td>
</tr>
<tr>
<td></td>
<td>Non-iid-3</td>
<td>0.0409</td>
<td>0.398</td>
<td>0.394</td>
<td>0.660</td>
<td>0.171</td>
<td>0.692</td>
</tr>
<tr>
<td></td>
<td>iid</td>
<td>0.1</td>
<td>0.697</td>
<td>0.694</td>
<td>0.710</td>
<td>0.120</td>
<td>0.715</td>
</tr>
<tr>
<td>Sign flipping</td>
<td>Non-iid-1</td>
<td>0.1</td>
<td>0.101</td>
<td>0.1</td>
<td>0.177</td>
<td>0.1</td>
<td>0.426</td>
</tr>
<tr>
<td></td>
<td>Non-iid-2</td>
<td>0.1</td>
<td>0.192</td>
<td>0.214</td>
<td>0.5131</td>
<td>0.1</td>
<td>0.621</td>
</tr>
<tr>
<td></td>
<td>Non-iid-3</td>
<td>0.1</td>
<td>0.397</td>
<td>0.398</td>
<td>0.651</td>
<td>0.1</td>
<td>0.686</td>
</tr>
<tr>
<td></td>
<td>iid</td>
<td>0.1</td>
<td>0.697</td>
<td>0.703</td>
<td>0.711</td>
<td>0.1</td>
<td>0.718</td>
</tr>
</tbody>
</table>

**Table 3.** The loss of the six defenses under four different data distributions on Cifar10, against three attacks.

<table>
<thead>
<tr>
<th>Attacks</th>
<th>Defenses</th>
<th>No</th>
<th>Krum</th>
<th>GeoMed</th>
<th>Abnormal</th>
<th>TrimmedMean</th>
<th>BRCA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Same value</td>
<td>Non-iid-1</td>
<td>2.84e16</td>
<td>11.72</td>
<td>9.61</td>
<td>2.29</td>
<td>6.05e17</td>
<td>2.09</td>
</tr>
<tr>
<td></td>
<td>Non-iid-2</td>
<td>6.99e16</td>
<td>7.29</td>
<td>8.01</td>
<td>2.06</td>
<td>3.63e16</td>
<td>2.09</td>
</tr>
<tr>
<td></td>
<td>Non-iid-3</td>
<td>4.48e16</td>
<td>2.35</td>
<td>2.38</td>
<td>1.893</td>
<td>3.37e16</td>
<td>0.691</td>
</tr>
<tr>
<td></td>
<td>iid</td>
<td>1.51e16</td>
<td>0.794</td>
<td>0.774</td>
<td>1.837</td>
<td>3.17e16</td>
<td>1.79</td>
</tr>
<tr>
<td>Gaussian noisy</td>
<td>Non-iid-1</td>
<td>8.635e4</td>
<td>8.41</td>
<td>9.37</td>
<td>2.29</td>
<td>936.17</td>
<td>1.54</td>
</tr>
<tr>
<td></td>
<td>Non-iid-2</td>
<td>9.51</td>
<td>7.57</td>
<td>8.37</td>
<td>1.34</td>
<td>7.98</td>
<td>0.623</td>
</tr>
<tr>
<td></td>
<td>Non-iid-3</td>
<td>8.22</td>
<td>2.01</td>
<td>2.31</td>
<td>0.94</td>
<td>6.07</td>
<td>0.692</td>
</tr>
<tr>
<td></td>
<td>iid</td>
<td>8.09</td>
<td>0.81</td>
<td>0.79</td>
<td>0.82</td>
<td>3.12</td>
<td>0.76</td>
</tr>
<tr>
<td>Sign flipping</td>
<td>Non-iid-1</td>
<td>2.30</td>
<td>10.72</td>
<td>9.91</td>
<td>2.29</td>
<td>2.30</td>
<td>1.54</td>
</tr>
<tr>
<td></td>
<td>Non-iid-2</td>
<td>2.31</td>
<td>7.77</td>
<td>7.10</td>
<td>1.34</td>
<td>2.30</td>
<td>0.621</td>
</tr>
<tr>
<td></td>
<td>Non-iid-3</td>
<td>2.31</td>
<td>2.36</td>
<td>2.13</td>
<td>0.94</td>
<td>2.30</td>
<td>0.686</td>
</tr>
<tr>
<td></td>
<td>iid</td>
<td>2.31</td>
<td>0.79</td>
<td>0.80</td>
<td>0.81</td>
<td>2.31</td>
<td>0.76</td>
</tr>
</tbody>
</table>
Our analysis is as follows: 1) The non-iid of data among clients causes large differences between clients’ models. And it is difficult for the defense method to judge whether the anomaly is caused by the non-iid of the data or by the Byzantine attacks, which increases the difficulty of defending the Byzantine attack. 2) Krum and GeoMed use statistical knowledge to select the median or individual client’s model to represent the global model. This type of method can effectively defend against Byzantine attacks when the data is iid. However when the data is non-iid, each client’s model only focuses on a smaller area, and its independence is high, cannot cover the domain of other clients, and obviously cannot represent the global model. 3) Trimmed Mean is based on the idea of averaging to defend against Byzantine attacks. When the parameter dimension of the model is low, it has a good performance. But as the complexity of the model increases, the method can not stably complete convergence.

5. Conclusions

In this work, we propose a robust federated learning framework against Byzantine attacks when the data is non-iid. BRCA detects Byzantine attacks by credibility assessment. Meanwhile, it makes the unified updating of the global model on the shared data, so that the global model has a consistent direction and its performance is improved. BRCA can make the global model converge very well when facing different data distributions. And for the pre-training of anomaly detection models, transfer learning can help the anomaly detection model get rid of its dependence on the test data set. Experiments have proved that BRCA performs well both on non-iid and iid data, especially on non-iid data. In the future, we will improve our methods by studying how to protect the privacy and security of shared data.

Acknowledgments

This work was partially supported by the Shanghai Science and Technology Innovation Action Plan under Grant 19511101300.

Conflict of interest

All authors declare no conflicts of interest in this paper.

References


©2022 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0)